3D-CE5.h: Merge candidate list for disparity compensated prediction

Thomas Guionnet, Laurent Guillo, Christine Guillemot

Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11
2nd Meeting: Shanghai, CN, 13–19 Oct. 2012

Document: JCT3V-B0080

Title: **CE5.h: Merge candidate list for disparity compensated prediction**

Status: Input Document

Purpose: Proposal

Author(s) or Contact(s): Thomas Guionnet Email: Thomas.Guionnet@inria.fr
Laurent Guillo Laurent.Guillo@inria.fr
Christine Guillemot Christine.Guillemot@inria.fr

Source: INRIA, France

Abstract

HEVC implements a candidate vector list for merge and skip modes. The construction of this list has been extensively studied in the JCT-VC group (see for instance JCTVC-G039). It has been shown in JCTVC-I0293 that it is possible to improve the HEVC coding performance by adding in the merge list copies of the first candidate shifted by an arbitrary offset. The same basis is considered in this document and applied to disparity compensation. A gain of 0.3 % is obtained on average on side views.

1 Introduction

HEVC relies on a candidate vector list for merge and skip modes. The efficiency of this approach depends highly on the relevance of the vectors present in the list. The construction of this list has been extensively studied [1]. During the 9th JCT-VC meeting, it has been shown in [2] that it is possible to improve the HEVC coding performance by adding in the merge list copies of the first candidate shifted by an arbitrary offset. More precisely, in [2], if there is available space in the candidate vector list, up to two new candidates are added: the first vector of the list shifted by an offset of (-4; 0) and (4; 0). It corresponds to a horizontal shift of -1 and 1 pixel respectively. In the context of video encoding of multiple rectified views, having a fine horizontal adjustment may be desirable for efficient disparity compensated prediction. Therefore, it is proposed to replace some candidates in the merge list with candidates pointing to the base view and shifted by horizontal offsets.

This document is the follow up of JCT3V-A0134 [3], as part of the core experiment CE5.h. It provides an updated description of the proposed method and results of the integration into HTM 4.0.1. A cross-check is available in JCT3V-B0098.

2 Proposed method

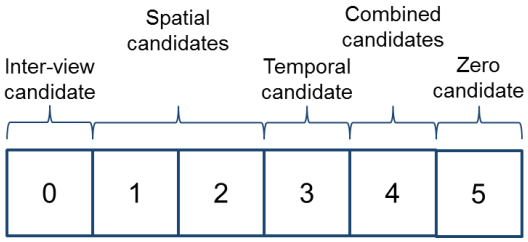

Figure 1: Candidate vector list for merge and skip mode example.

Figure 1 depicts a typical example of motion vector candidate list. Each motion vector candidate is defined as

- the luma motion vectors mvL0 and mvL1,
- the reference indices refIdxL0 and refIdxL1,
- the prediction list utilization flags predFlagL0 and predFlagL1.

The candidate list construction proceeds as follows.

1. Construct the list as currently specified (example result depicted on Figure 1).
2. **Scan the list from the first to the fourth position**. The first candidate found in the list during this scan which has either its L0 vector or L1 vector or both pointing to another view is marked as *selected candidate* (see for example Figure 2). A vector mvLX is pointing to another view if the viewId of current coding unit is not equal to the viewId of reference refIdxLX. **If no such candidate exists, the process ends.**
3. Let *m* be the index of the selected candidate. The index *m* is greater or equal to 0 and lower or equal to 3. The selected candidate is defined by $mvL0_m$, $mvL1_m$, $refIdxL0_m$, $refIdxL1_m$, $predFlagL0_m$, and $predFlagL1_m$. Two refined candidates candDCPa and candDCPb are constructed as follows:
    a. **If** $mvL0_m$ is pointing to another view:
        - $mvL0candDCPa = mvL0_m + (4, 0)$
        - $mvL1candDCPa = mvL1_m$
        - $refIdxL0candDCPa = refIdxL0_m$
        - $refIdxL1candDCPa = refIdxL1_m$
        - $predFlagL0candDCPa = predFlagL0_m$
        - $predFlagL1candDCPa = predFlagL1_m$

        - $mvL0candDCPb = mvL0_m + (- 4, 0)$
        - $mvL1candDCPb = mvL1_m$
        - $refIdxL0candDCPb = refIdxL0_m$
        - $refIdxL1candDCPb = refIdxL1_m$
        - $predFlagL0candDCPb = predFlagL0_m$
        - $predFlagL1candDCPb = predFlagL1_m$

    b. **Else** ($mvL1_m$ vector is pointing to another view):
        - $mvL0candDCPa = mvL0_m$
        - $mvL1candDCPa = mvL1_m + (4, 0)$
        - $refIdxL0candDCPa = refIdxL0_m$
        - $refIdxL1candDCPa = refIdxL1_m$
        - $predFlagL0candDCPa = predFlagL0_m$

- predFlagL1candDCPa = $predFlagL1_m$

- mvL0candDCPb = $mvL0_m$
- mvL1candDCPb = $mvL1_m$ + (- 4, 0)
- refIdxL0candDCPb = $refIdxL0_m$
- refIdxL1candDCPb = $refIdxL1_m$
- predFlagL0candDCPb = $predFlagL0_m$
- predFlagL1candDCPb = $predFlagL1_m$

4. Write the two refined candidates in the list as follows:
   a. **If** $m < 3$ (illustrated by Figure 3 and Figure 4)
      - $mvL0_5$ = $mvL0_3$
      - $mvL1_5$ = $mvL1_3$
      - $refIdxL0_5$ = $refIdxL0_3$
      - $refIdxL1_5$ = $refIdxL1_3$
      - $predFlagL0_5$ = $predFlagL0_3$
      - $predFlagL1_5$ = $predFlagL1_3$

      - $mvL0_3$ = mvL0candDCPa
      - $mvL1_3$ = mvL1candDCPa
      - $refIdxL0_3$ = refIdxL0candDCPa
      - $refIdxL1_3$ = refIdxL1candDCPa
      - $predFlagL0_3$ = predFlagL0candDCPa
      - $predFlagL1_3$ = predFlagL1candDCPa

      - $mvL0_4$ = mvL0candDCPb
      - $mvL1_4$ = mvL1candDCPb
      - $refIdxL0_4$ = refIdxL0candDCPb
      - $refIdxL1_4$ = refIdxL1candDCPb
      - $predFlagL0_4$ = predFlagL0candDCPb
      - $predFlagL1_4$ = predFlagL1candDCPb

   b. **Else** ($m = 3$, illustrated by Figure 5)
      - $mvL0_4$ = mvL0candDCPa
      - $mvL1_4$ = mvL1candDCPa

- $refIdxL0_4$ = refIdxL0candDCPa
- $refIdxL1_4$ = refIdxL1candDCPa
- $predFlagL0_4$ = predFlagL0candDCPa
- $predFlagL1_4$ = predFlagL1candDCPa

- $mvL0_5$ = mvL0candDCPb
- $mvL1_5$ = mvL1candDCPb
- $refIdxL0_5$ = refIdxL0candDCPb
- $refIdxL1_5$ = refIdxL1candDCPb
- $predFlagL0_5$ = predFlagL0candDCPb
- $predFlagL1_5$ = predFlagL1candDCPb

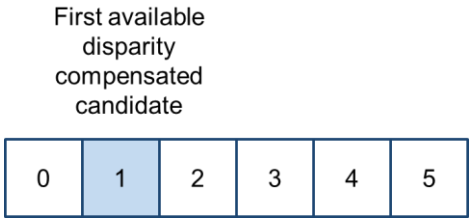

Figure 2: Selection of a disparity compensated candidate.

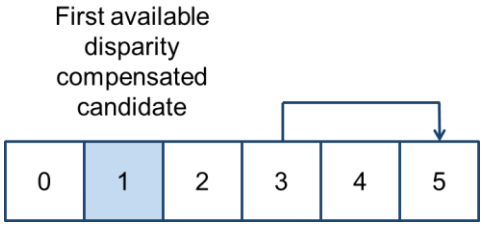

Figure 3: Displacement of the fourth candidate to the last position in the candidate list.

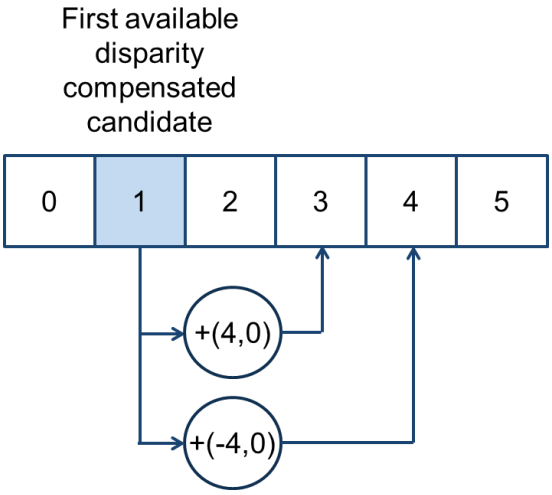

Figure 4: Overwriting of fourth and fifth candidates by newly generated candidates.

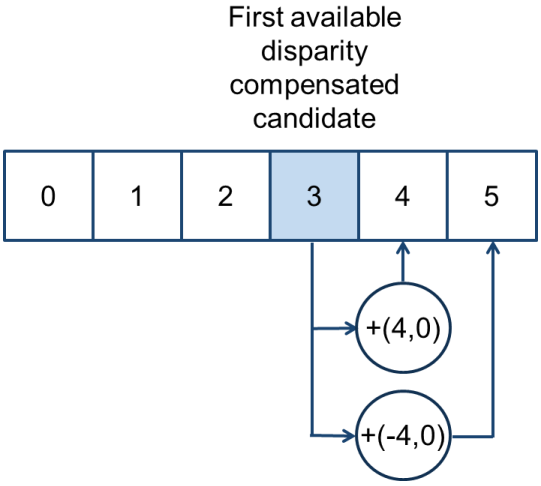

Figure 5: Overwriting of fifth and last candidates by newly generated candidates

3 Experimental results

The experiment has been conducted with HTM 4.0.1 and evaluation is based on common test conditions [4].

Results are provided in Table 1.

Table 1: Results obtained with HTM 4.0.1 and Merge list extension with DCP candidate, compared to HTM 4.0.1

	video 0	video 1	video 2	video only	synthesized only	coded & synthesized	enc time	dec time
Balloons	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	96,8%	101,1%
Kendo	0,0%	-0,2%	-0,3%	-0,1%	0,0%	-0,1%	97,0%	101,2%
Newspapercc	0,0%	-0,5%	-0,5%	-0,2%	0,0%	-0,1%	96,9%	99,0%
GhostTownFly	0,0%	-1,1%	-0,8%	-0,3%	-0,2%	-0,2%	97,3%	101,5%
PoznanHall2	0,0%	0,2%	-0,3%	-0,1%	-0,1%	-0,1%	97,9%	99,2%
PoznanStreet	0,0%	-0,5%	-0,3%	-0,2%	-0,1%	-0,2%	95,6%	100,5%
UndoDancer	0,0%	-0,3%	-0,2%	-0,1%	-0,1%	-0,1%	96,9%	100,9%
1024x768	0,0%	-0,2%	-0,3%	-0,1%	0,0%	0,0%	96,9%	100,4%
1920x1088	0,0%	-0,4%	-0,4%	-0,2%	-0,1%	-0,1%	96,9%	100,5%
average	**0,0%**	**-0,3%**	**-0,3%**	**-0,1%**	**-0,1%**	**-0,1%**	**96,9%**	**100,5%**

4 Conclusion

Given the reported gains with no added complexity, we suggest to adopt the proposed method.

5 References

[1] Benjamin Bross, Joel Jung, "CE9: Summary Report of Core Experiment on MV Coding and Skip/Merge Operations", JCTVC-G039, JCT-VC 7th Meeting, Geneva, CH, 21-30 November, 2011.

[2] Tomoyuki Yamamoto, Tomohiro Ikai, "Merge candidate refinement for uni-predictive block", JCTVC-I0293, JCT-VC 9th meeting, Geneva, CH, 27 April – 7 May 2012.

[3] Thomas Guionnet, Laurent Guillo, Christine Guillemot, « CE5.h related: Merge candidate list extension for disparity compensated prediction», JCT3V-A0134, July 2012, Stockholm, Sweden.

[4] Dmytro Rusanovskyy, Karsten Müller, Anthony Vetro, « Common test conditions for 3DV Core Experiments », JCT3V-A1100, July 2012, Stockholm, Sweden.

6 Patent rights declaration(s)

INRIA may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).